REMARKS

Claims 1, 12 and 17 have been amended. Upon entry of this amendment, claims 1-20 remain pending. Some minor amendments to insert term "providing" at two locations in claim 1 have been made to improve the readability of claim 1 without changing its scope.

Basis for Claim Amendments

Basis for the remaining amendments to claims 1, 12 and 17 can be found on page 5, lines 8-17, and, in particular, page 5, lines 8-11 of the application as originally filed. This paragraph makes it clear that the execution of initiated tasks is prevented if it would have been brought about by the detected command. This passages at page 8, line 34 to page 9, line 3 and page 9, line 19 of the application as originally filed, indicate that a system administration program detects a command to execute a task.

The Drawings

The Examiner's indication that the drawings have been accepted is hereby acknowledged with appreciation.

The Rejection Under 35 U.S.C. §103(a)

Claims 1-20 have been rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent no. 5,689,708 (hereinafter "Regnier") in view of U.S. Patent no. 6,513,111 (hereinafter "Klimczak"). This rejection, at least insofar as it applies to claims 1-20, as amended, is traversed and reconsideration is requested for the reasons which follow.

Regnier describes the control of system resources in computer system arranged in a client/server configuration. Resource restrictions can be imposed in a way which is transparent to the user, through the cooperation of a profile for each user and a control facility in each application program, which can read the current user's profile in order to manage a specified set of system resources (column 3, lines 22-26 of Regnier). Each of the user profiles may be different (column 3, lines 22-26 of Regnier).

The method of claim 1 of the present application differs from the method of Regnier, in that Regnier does not disclose:

(1) providing a task database of tasks,

(2) running a system administration program to configure a user-specific list of allowed tasks on the basis of the user profile in the user database and the task database, or

(3) preventing execution of tasks of which the execution is initiated by a command detected by a system administration program, on determining that such tasks are not on the list of allowed tasks configured by a system administration program.

Instead of running a system administration program to configure a user-specific list of allowed tasks on the basis of the user profile in the user database and a task database, Regnier teaches that user profiles are generated and stored in a server by a system administrator (column 3, lines 44-45 of Regnier). The network or system administrator decides which users are allowed to employ which applications, and which users are allowed to access which systems resources (column 6, lines 23-26 Regnier). The user profiles are comprised in a database table 400 that lists a user or predefined group of users, the name of a particular application program, a resource potentially used by that application program, a value showing a status of that resource for that particular user when executing that particular application program, and a change status for the value (column 8, lines 14-20 of Regnier). Thus, the user profiles described in Regnier comprise a user-specific list of tasks generated and stored by a system administrator, and they are not configured by running a system administration program.

Accordingly, it is clear that the following elements of claim 1 are not disclosed in Regnier:

(1) providing a task database of tasks, and

(2) running a system administration program to configure a user-specific list of allowed tasks on the basis of the user profile in the user database and the task database.

Further, since Regnier does not run a system administration program to configure a user-specific list of allowed tasks, the following additional limitation of claim 1 is cannot be disclosed in Regnier since it requires reference to a user-specific list of tasks configured by a system administration program:

(3) preventing execution of tasks of which the execution is initiated by a command detected by a system administration program, on determining that such tasks are not on the list of allowed tasks configured by a system administration program.

Regnier does not teach preventing execution of tasks of which the execution is initiated by a command detected by a system administration program, on determining that such tasks are not on the list of allowed tasks. In one aspect, Regnier teaches that restrictions are enforced by changing the application program's interface to the user under control of the user's profile (column 3, lines 32-35 of Regnier). When a user executes a particular application program, the application program itself governs which system resources are available to the user, independently of the operating system or other programs running on the client or on the server (column 3, lines 50-53 of Regnier).

When a user selects an application, block 307 in the resource manager client-module of Fig. 3 of Regnier determines whether or not that application program complies with the invention of Regnier (column 6, lines 48-53 and Fig. 3 of Regnier). If the application program does not comply with the invention of Regnier, the application is still executed in the conventional manner (col. 6, lines 56-58 of Regnier).

Regnier does disclose that the application's menu choices, i.e. the list of facilities which it presents to the user, may be changed to eliminate forbidden menu choices (col. 3, lines 49-52 of Regnier). Also, Regnier contemplates preventing execution of applications by eliminating or dimming menu choices for such applications based on the specific user profile so that the user could not choose them in the first place. (col. 7, lines 10-14 of Regnier). Thus, since, in this embodiment, forbidden choices cannot be selected by the user, Regnier does not prevent execution of tasks or applications initiated by a detected command, since it is not possible for such a command to be provided by the user because the forbidden commands cannot be selected in the first place. Execution of these tasks is not initiated by the command that is provided, namely the command selecting an application. Instead, execution of these tasks is prevented by preventing issuance of the command in the first place.

Regnier also mentions that in a case that an application is requested by the user and execution of the program is not allowed, block 311 returns control to block 307 for selection of another application. See col. 7, lines 8-10 of Regnier. However, Regnier

does not indicate that block 311 of Fig. 3 consults a user-specific list of allowed tasks configured by a system administration program to determine if execution of the task is allowed, as required by the present claims. Rather, the only indication given in Regnier as to how block 311 of Fig. 3 may function is that if authentication of a program by block 310 of Fig. 3 is proper, block 310 transmits a positive response to block 311 (See col. 7, lines 5-7 of Regnier). Authentication is performed to prevent application spoofing or to prevent an application from claiming resources that the application is not entitled to. (Col. 6, line 66, to col. 7, line 2 of Regnier). Thus, there is no mention of either block 310 or block 311 of Fig. 3 of Regnier consulting a user-specific list of allowed tasks configured by a system administration program to determine whether execution of a particular task is allowed.

The effect of the differences identified above is that the invention allows for efficient provision of flexible user-specific lists of allowed tasks and prevents other tasks from being executed, including unknown other tasks. Because a database of tasks and a user database, comprising a user profile for each user, is provided, and the user-specific list of tasks is configured on the basis of the user profile in the user database and the task database, the user-specific list of tasks can be flexible. Because a system administration program is run to generate such lists, provision of the list of tasks is carried out efficiently. Because a system administration program detects a command to execute a task, and the execution of tasks of which the execution is initiated by the detected command is prevented on determining that they are not on the list of allowed tasks, and the list of allowed tasks comprises allowed application programs, the method is usable to prevent code from unknown sources from being executed. This includes, for example, programs downloaded or installed by a user himself, unbeknownst to a system administrator.

Contrary to the position of the Examiner, the skilled person would not seek to combine Regnier with Klimczak in order to prevent execution of applications, because Klimczak relates to providing customized access to software applications, involving the configuration of action items within the software (see column 1, lines 13-15 and 52-55), and not to preventing the applications, including unknown other tasks, from being executed.

Moreover, the systems in Regnier and Klimczak are incompatible, since Regnier requires that server module 240, interact with specially adapted applications, whereas Klimczak discloses software that can be set up by a subscriber (column 3, lines 21-23 of Klimczak). In other words, Klimczak discloses an application program that can be customized.

The invention could not be arrived at by combining Regnier and Klimczak, because Klimczak does not disclose preventing execution of tasks of which the execution is initiated by a command detected by a system administration program, on determining that they are not on the list of allowed tasks, nor does Klimczak teach the provision of a list of allowed tasks comprising allowed application programs.

Although the "action items" described in Klimczak might conceivably be regarded as tasks in some implementations, they are not equivalent to application programs. Instead, according to Klimczak, an entitlement agent returns user profile information to a requesting object (column 11, lines 50-52). When a module is started, user profile information is requested from a user profile database (column 6, lines 56-57). The functionality of the software will then be controlled for the user in accordance with the value of the action item for that user (column 12, lines 57-60). The method of Klimczak does not work for software objects that are not part of the described suite, and therefore not able to generate .PRO files as described in the passage running from column 9, line 26 to column 10, line 46 of Klimczak. Thus, there is no disclosure in Klimczak of a list of allowed tasks comprising allowed application programs.

The Examiner's reasoning that a task is always equivalent to an application program cannot be followed. Although an application program is a particular species of task in the definition of the present application, the reverse is not true. Thus, a task is not a species of an application program. This means that, according to the definition in the specification:

(1) an application program is always a task, and

(2) a task may or may not be an application program, since tasks include other things that are not application programs.

The feature of providing a list of allowed tasks comprising allowed application programs of the present claims should be read in conjunction with the feature that a

system administration program detects a command to execute a task, wherein the execution of tasks of which the execution is initiated by the detected command is prevented if they are not on the list of allowed tasks. In that way, importation and subsequent running of unknown application programs is prevented since the list of tasks comprises a list of allowed application programs. Such features are not known from Klimczak.

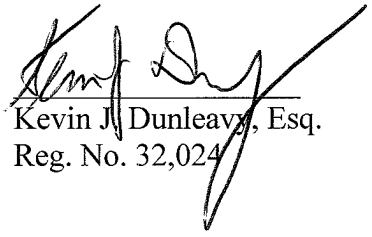
Instead, by customizing an interface (column 12, lines 26-29), the method of Klimczak prevents the user from issuing a very particular set of commands to execute a very particular set of known tasks. Preventing a command from being issued is different from preventing execution of a task for which the execution has already been initiated by a detected command.

Of the other cited publications, U.S. Patent no. 6,339,826 describes a method wherein a server uses a system identifier and password to build a list of applications to which the user has access permission, with the application list being used to build a portion of the desktop of applications to which the user has access permission (column 4, lines 36-39). U.S. Patent no. 6,401,238 describes resolving a rule against monitored conditions and a user profile to select an application version of a managed application to serve to a client computer (column 8, lines 56-59). U.S. Patent no. 6,546,002 describes a mobile interface on a local device, which mobile interface includes a plurality of pointers corresponding to user specific resources and information (column 17, lines 14-18).

Thus, the prior art consists of methods whereby the user's interface to a pre-defined set of application programs is restricted. It provides no solution to the problem of attempts by users to run unknown application programs in the case where they system permits the execution of a command to run such application programs. The present invention allows only tasks on a user-specific list of allowed tasks, comprising allowed application programs, to be executed, because a system administration program detects commands leading to the initiation of execution of other tasks. Because the user-specific list of allowed tasks can be provided efficiently and flexibly, the stringent control of the user's list of allowed tasks does not impose an appreciable burden on either system administrators or users. Such an advantageous effect is not achieved by any of the methods described in the state of the art cited by the Examiner.

In view of the above amendments and the foregoing remarks, Applicant respectfully submits that all of the pending claims are in condition for allowance and respectfully request a favorable Office Action so indicating.

Respectfully submitted,



Date: August 17, 2006

Kevin J. Dunleavy, Esq.
Reg. No. 32,024

Customer No. 21302
KNOBLE YOSHIDA & DUNLEAVY, LLC
Eight Penn Center, Suite 1350
1628 John F. Kennedy Blvd.
Philadelphia, PA 19103
Tel: (215) 599-0600
Fax: (215) 599-0601